BABEL SPEEDS COMMUNICATION

BABEL is the Hebrew name for the ancient city Babylon, where, according to the Bible, construction on an enormous tower ended in confusion because the people working on the project spoke different languages. In a similar manner, building a scientific application is often complicated by the many programming languages used. Making these codes communicate with each other can require expensive and labor-intensive programming.

A new Lawrence Livermore program called Babel makes sense of the confusion. Developed with funding from Livermore's Laboratory Directed Research and Development Program by a team of Livermore computer scientists, Babel earned a 2006 R&D 100 Award for being the world's most rapid means of communicating among many programming languages. With this application, applications in different languages can interoperate, allowing them to pass scientific data seamlessly and efficiently from one to another. The user community, which includes scientists from almost 20 fields, can combine programs with minimal loss in performance. What's more, Babel facilitates access to outstanding applications that were formerly too difficult or expensive to integrate manually.

Targeting Massively Parallel Supercomputers

Although Babel is a general-purpose tool, its primary market is high-performance, scientific software running on massively parallel supercomputers. "Scientific applications are becoming more complex, incorporating more types of physics, numerical methods, and visualization techniques," says Livermore computer scientist Thomas Epperly, who leads the Babel project. Computational scientists developing new applications face difficulties in combining software packages because different languages such as Python, C++, and Fortran are used to write them.

Epperly, a member of Livermore's Center for Applied Scientific Computing (CASC), explains that some combinations can be connected with relative ease—for example, a C++ package that calls, or interoperates with, a C function. For other pairs, such as C++ and Fortran 95, developers must write "glue code" to connect the pieces. In the worst case, computational scientists may need to rewrite a particular library from scratch or not use the library at all.

The Babel toolkit reduces this manual programming burden by automatically generating the glue code. "Babel provides a consistent, semi-automated approach for coupling software pieces written in different programming languages," says Epperly.

The program manages all of the function calls between any combination of C, C++, Fortran 77, Fortran 90/95, Java, and Python, while hiding the complex operations from the user. Babel automatically determines the interoperability issues and then generates the software needed to translate data between the requested languages. From a programmer's viewpoint, calling between languages is seamless. Another advantage for the developer is the consistent interface used to set up these connections.

Working with Babel, scientists can integrate almost any library or third-party tool into a scientific application. Epperly says Babel ends the "programming language wars" because the best language for each component can be used. "Programmers can work in their favorite language," he says. "No one needs to be disenfranchised."

INTRODUCTION TO FORTRAN
The C++ Programming Language
Stroustrup
Fortran 90
The Java Programming Language
THE C PROGRAMMING LANGUAGE
DEC Fortran Language Reference Manual
Python Essential Reference
Python Essential Reference
BABEL
Winner 2006 R&D 100
v1.0
www.llnl.gov/CASC/components/
components@llnl.gov

The Livermore-developed program called Babel allows software applications in different programming languages to communicate quickly. It is available at: www.llnl.gov/CASC/components.

AMONG PROGRAMMING LANGUAGES

Strong Support for Fortran

One of Babel's strongest features is its support for Fortran, a language that similar tools ignore. "Many programmers think Fortran is a dead language from the early days of computing," says Epperly. "Although largely ignored by mainstream computing, it is still very much alive in scientific and engineering computing." Babel currently supports Fortran 77 and 90/95 and will support Fortran 2003 in the near future. "The language-interoperability technologies available on the market address only a small subset of languages that don't include Fortran," says Epperly. "Those tools are useful for many operations, but they do not solve the general interoperability problem as more languages are mixed into a single application."

Babel supports the complex numbers and multidimensional arrays that are common in scientific applications. It also allows applications to transfer array data between languages without having to make copies of the data in the new language, which can significantly decrease performance.

In developing Babel, the Livermore team designed the Scientific Interface Definition Language (SIDL) to address the unique needs of parallel scientific computing. SIDL provides a road map for transferring data between languages.

Babel is used increasingly by researchers who are developing common standards for high-performance computing. Livermore is a member of the Common Component Architecture Forum, which includes national laboratories and academic institutions committed to defining a component standard for high-performance computing. Such a standard will promote interoperability between components developed by teams at different institutions. Epperly explains that community standard interfaces are valuable because they allow developers to choose between several software libraries without having to modify existing code. In addition, ancillary tools will work with all of the code libraries that meet the community standard.

Babel is available at no cost from the CASC Web site at www.llnl.gov/CASC/components and is downloaded on average about 90 times per month. Users also have access to online documentation, reports, and frequently asked questions. Babel's current customers are programmers at national laboratories and universities. These scientists are developing advanced applications in diverse disciplines including astronomy, biological and medical simulations, climate and weather modeling, climate feature extraction, computational chemistry, data management, fusion and plasma physics modeling, multiscale materials science, nanophotonics, nuclear power plant simulations, performance measurement, and scientific visualization.

With Babel's help, the supercomputing world can readily support many different languages, all which can work together to facilitate scientific discovery.

—Arnie Heller

Key Words: Babel, Common Component Architecture Forum, Fortran, language interoperability, R&D 100 Award, Scientific Interface Definition Language (SIDL).

For further information contact Thomas Epperly (925) 424-3159 (epperly2@llnl.gov).

The Babel development team (from left to right): James Leek, Thomas Epperly, Tamara Dahlgren, Scott Kohn, and Gary Kumfert.